

VIDEO ARTIFACT IDENTIFICATION AND COUNTING

Field of the Invention

The present invention is directed to video signal processing and, more particularly, to artifact detection in video signal processing.

Background

Computer arrangements, including microprocessors and digital signal processors, have been designed for a wide range of applications and have been used in virtually every industry. For a variety of reasons, many of these applications have been directed to processing video data and have demanded minimal levels of power consumption and compactness. Some applications have further demanded a high-speed computing engine that can perform effectively on a real-time or near real-time basis. Many of these video-processing applications have required a data-signal processing circuit that is capable of performing multiple functions at ever-increasing speeds.

Various video applications, such as moving pictures experts groups (MPEG) applications, MPEG2 applications (*e.g.*, all-digital transmission of video data at coded bitrates between about 4 and 9 Mbit/sec) and others employ compressed video data. Data compression is particularly useful in data processing and transfer because smaller amounts of data can typically be transferred faster and use a smaller bandwidth. However, data compression can sometimes produce errors resulting from lost data. The lost data typically shows up as a discrepancy, or artifact, in the video image. As data is compressed smaller, the occurrence of artifacts increases. In this regard, there is a tradeoff between increased compression and increased artifacts that occur as a result of the compression.

Video images created using compressed data may include several artifacts without significantly affecting image quality or at least without reducing the image quality below a particular level of acceptability. In certain video processing applications, an acceptable count, or number, of artifacts is selected and the compression of video data is controlled so that the artifact count is maintained within an acceptable range. In these applications, it is generally

useful to count and/or identify the artifacts for taking corrective or other action in response to the count.

In video sequences that use MPEG2 compressed and decompressed data, several types of artifacts can be found. These sequences typically employ an encoding format such as 8x8 (MPEG2), 8x10 (540 or 528 pixels), or 8x12 (480 pixels) formats. Due to problems related to issues such as quantization of 8x8 block discrete cosine transform (DCT) coefficients or poor encoded results, artifacts can be amplified through processes such as contrast or sharpness enhancement. In these and other instances, noticeable blocking artifacts can be identified and are commonly referred to in connection with the blockiness of a particular video sequence. Blocking artifacts commonly show up momentarily as artificial rectangular discontinuities in a decompressed decoded image. The visibility of the blocking artifacts generally depends on the amount of compression used, the quality and nature of the original pictures as well as the quality of the coder used. The visible blocks may include 8 x 8 DCT blocks or misplaced blocks (e.g., 16 x 16 pixel macroblocks) due to the failure of motion prediction and/or estimation in an MPEG coder or other motion vector system, such as a standards converter. It is desirable to take corrective actions to reduce the visibility of these artifacts.

The present invention is directed to goals including the above-mentioned and the detection of artifacts in MPEG2 and other types of video data processing.

Summary of the Invention

Various aspects of the present invention are directed to artifact detection in video data processing, and in a more specific application to artifact identification and counting.

Consistent with one specific example embodiment, the present invention is directed to artifact detection and counting that includes the use of two loops in each of a horizontal and vertical direction in a data table to accumulate a count of detected artifacts. The loops are effected in a manner that addresses challenges, including those discussed above, related to artifact detection and video data management.

The above summary of the present invention is not intended to describe each illustrated embodiment or every implementation of the present invention. The figures and the detailed description that follow more particularly exemplify these embodiments.

Brief Description of the Drawings

The invention may be more completely understood in consideration of the following detailed description of various embodiments of the invention in connection with the accompanying drawings, in which:

FIG. 1 shows artifact detection and counting, according to an example embodiment of the present invention;

FIG. 2 is a horizontal grid for artifact counting, according to another example embodiment of the present invention; and

FIG. 3 is a vertical grid for artifact counting, according to another example embodiment of the present invention.

While the invention is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the invention to the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

Detailed Description of Various Example Embodiments

The present invention is believed to be highly applicable to methods and arrangements for artifact detection in video data processing, such as in MPEG2 processing. The invention has been found to be particularly advantageous for video data processing using data compression and that is susceptible to artifacts. While the present invention is not necessarily limited to such applications, an appreciation of various aspects of the invention is best gained through a discussion of examples in such an environment.

In one example embodiment of the present invention, artifacts in a video data sequence are detected and counted using a two-part algorithm employing two scanning loops in each of the horizontal and vertical directions to accumulate artifact count values in an artifact count table. The count table is updated (increased) when an unwanted artifact, such as contouring, is detected for either a horizontal or vertical artifact. A first loop is scanned

across the table and uses quotient and remainder values of contour from a software application to increase entries in the count table, up to a maximum length of the remainder value. A second looping pass increases artifact entries in the table with the quotient value of contour, less one, and up to the end of the row or column of the table being looped through. The artifact detection is implemented using relatively low-cost hardware and exhibits relatively small bandwidth and latency.

The present invention may be implemented in connection with a variety of applications. FIG. 1 shows one such implementation in which the artifact counting of the present invention is employed. An MPEG2 artifacts detection unit is adapted to detect the existence of artifacts in video sequences using gradient tests in both horizontal and vertical directions. An output from the detection unit represents an indication of blockiness in scanning the active region of pixel processing and of the grid size and origin of artifacts. Scanned data 105 is processed in an MPEG2 artifacts detection algorithm consisting of pixel and field processing 120 and 140, respectively. The pixel processing is divided into gradient filtering and identification of blocking artifacts. The identification of artifacts is divided into three steps including testing, artifacts counting and grid size counting. The gradient filtering and identification of blocking artifacts is implemented in hardware. The field processing includes blockiness and grid size procedures. The blockiness procedures are used by block level indication and grid position processing, and the grid size is determined using grid count values from MMIO registers to perform the calculation for a new grid size. The field processing is implemented in software design.

The artifacts counting is based on test conditions of gradient values or neighboring pixels in horizontal or vertical directions. If the gradient condition is met, there is an artifact existing in the pixel. Horizontal (test_H) and vertical (test_V) artifact test values determine the increment of the count values within an artifact count table for horizontal and vertical testing. The output values for the artifacts count table are stored to memory and the grid count is stored to MMIO registers. The field processing is adapted to access the stored values through registers.

In horizontal artifact counting, the length of artifacts existing in the horizontal direction (contour_H) is defined as:

$$\text{contour_H} = \text{contour_H_Q} * \text{PREVIOUS_GRID_SELECT} + \text{contour_H_R},$$

where contour_H_Q is the quotient of the contour_H and PREVIOUS_GRID_SELECT, which is the predicted grid size (*e.g.*, 8, 10, or 12); and where contour_H_R is the remainder of the contour_H. Once the length of horizontal artifact is assigned, the counting bounds are determined as from [PREVIOUS_GRID_SELECT, contour_H]. Horizontal entries in the artifacts count table are not incremented when the counting value is out of bounds (*e.g.*, higher or lower than set bounds for the table). If the count value is within the range of bounds, the artifacts count table increments its entries until the decrement of count value reaches a lower bound.

In vertical artifact counting, the count value for vertical artifacts is incremented in a similar manner to that of horizontal artifact counting. The length artifacts existing in the vertical direction (contour_V) is defined as:

$$\text{contour_V} = \text{contour_V_Q} * 8 + \text{contour_V_R},$$

where contour_V_Q is the quotient of the contour_V and contour_V_R is the remainder. Once the length of the vertical artifact is set, the counting bounds are determined from [8, contour_V]. The artifacts count table does not increment its vertical entries when the counting value is out of bounds, and if the count value is within the range of bounds, the artifacts count table increments its entries until the decrement of count value reaches a lower bound.

The maximum count accumulation cycle time is completely independent of the contour values and the additions cycles are also reduced. By using the quotient and remainder of contour values from a software application, the artifacts identification and counting are reduced from a conventional [contour_H_R,, contour_H_Q * PREVIOUS_GRID_SELECT +

contour_H_R] to [contour_H_R, PREVIOUS_GRID_SELECT] for horizontal counting and from [contour_V_R, contour_V_Q * 8 + contour_V_R] in conventional applications to [contour_V_R, 8] vertical counting (where a vertical grid size of 8 is used).

In one particular example embodiment of the present invention, the horizontal count storing cycle of upper bound ratio is given as:

$$\frac{PREVIOUS_GRID_SELECT}{PREVIOUS_GRID_SELECT \cdot contour_H_Q + contour_H},$$

where PREVIOUS_GRID_SELECT is 8, 10, or 12. And the vertical count storing cycle time of upper bound ratio is given as:

$$\frac{8}{8 \cdot contour_V_Q + contour_V}.$$

The row or column entries for the table are divided into two groups for storing the values for incrementing (inc) or (inc-1). FIG. 2 shows example table sizes for the horizontal direction, with grid portions 310, 320 and 330 corresponding to previous grid sizes of 12, 10 and 8, respectively. Portions 312, 322 and 332 correspond to the inc-1 values, and portions 314, 324 and 334 correspond to the inc values. The increment value for horizontal direction is defined as:

$$inc = \frac{count_value}{previous\ select\ grid\ siz},$$

where the count value is the horizontal count value, and the boundary point is defined as:

$$boundary = count\ value \% previous\ select\ grid\ size.$$

FIG. 3 shows an example table size (8) with grid portion 310 for vertical artifacts storing, with portions 312 and 314 corresponding to inc-1 and inc values, respectively. The boundary is the remainder and the increment value for the vertical direction is defined as:

$$inc = \frac{count_value}{8},$$

and the boundary for vertical direction is also defined as:

$$boundary = count_value \% 8.$$

The original bandwidth for storing the count values into entries depends on the contour values. The maximum bandwidth for storing the count values can be accomplished with only processing one row or one column once. This is particularly useful for reducing the bandwidth required, as compared to conventional applications, where the contour values are set to large number. In addition, the boundary for two types of additions for entries can be automatically wrapped around within the artifacts count table.

For more information regarding uses and implementations of the present invention, reference may be made to the attached appendix, entitled "A new artifact identification and counting method for MPEG2 artifact detection in Quality Video Input," which is fully incorporated herein by reference.

The present invention should not be considered limited to the particular examples described above. Various modifications, equivalent processes, as well as numerous structures to which the present invention may be applicable fall within the scope of the present invention, as fairly set forth in the appended claims.